Refinement against cryo-EM data with CCP-EM using REFMAC & Coot

Written by Rob Nicholls, Tom Burnley, Colin Palmer, Keitaro Yamashita & Garib Murshudov

Tutors: Tom Burnley, Colin Palmer, Arjen Jakobi

NEMI Cryo-EM School, Delft, 2022

In this tutorial we consider the refinement of a beta-galactosidase model into 2.9 Å maps from a cryo-EM reconstruction from the Relion tutorial using a homologous model from the PDB as a starting point (homolog.pdb).

The complete biological assembly comprises four chains, however as these are symmetry mates we use Refmac-Servalcats symmtry functions and only require one chain. The homologous model used as the starting point corresponds to chain A from the deposited model with PDB code: 5a1a.

Part 1) Map and Model Preparation

`(a) Open Coot, and load the model (homolog.pdb) and map (postprocess.mrc)`

- For use with MacBooks without a mouse:
  - `X11 -> Preferences -> Emulate 3 Button mouse`

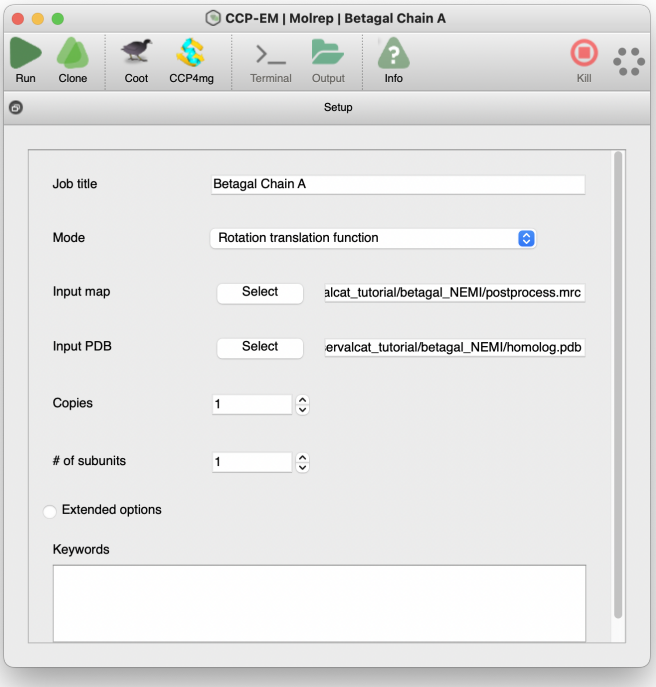
  This will simulate the pressing of the middle and right buttons when you use it in conjunction with Option and Command keys.
- Customise Coot for cryoEM:
  - `Calculate -> Modules -> Cryo-EM`
    - This will load an additional Cryo-EM menu
- Move the view to the centre of the density and sets the contour level
  - `Cryo-EM -> Go to Map Molecule Middle`

  If you don't see the map, but you do see it listed in the Display Manager, then it could be that the contour level is too high – try decreasing the map contour level (scroll down with the mouse, or press "-" a few times).
- It can be useful to display the box that encapsulates the map:
  - `Draw -> Cell & Symmetry -> Show Unit Cells? -> Yes`
- Then zoom in/out until you see the whole box (right click and drag, or use the "m" and "n" keys).
- Increase the map radius in order to display the whole map (e.g. to 70 Å in this case):
  - `Edit -> Map Parameters -> Map Radius EM.`

You'll notice that the model and the map aren't aligned…

`(b) Dock the model into the map`

The simplest way is to perform a rigid body rotation and translation search using the Molrep task in CCP-EM.



Start the Molrep in the CCP-EM main gui and

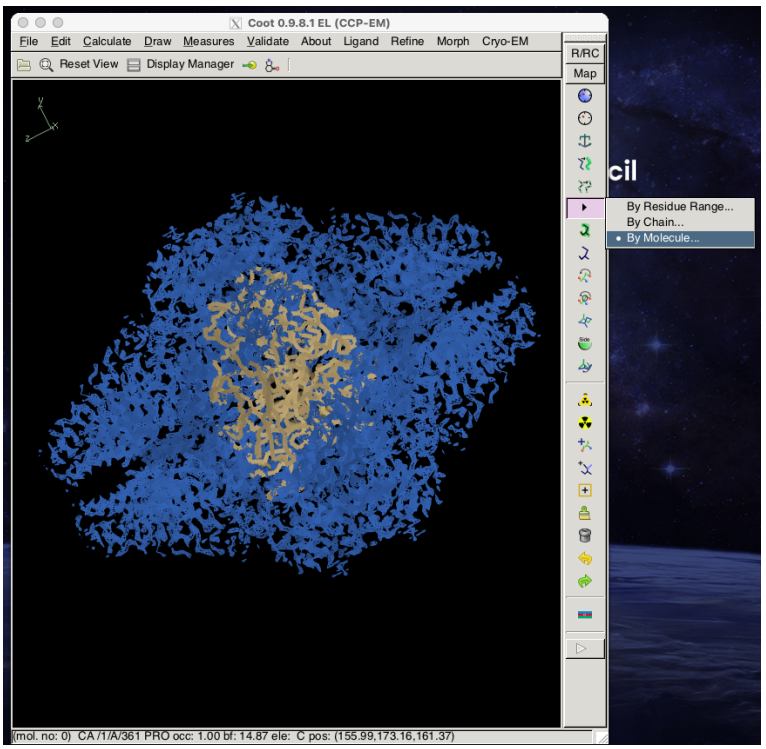
- Set the following parameters:
    - `Mode "Rotation translation function"`
    - `Input map: postprocess.mrc`
    - `Input pdb: homolog.pdb`
    - `Copies: 1`
- Press Run.

Note: "Mode" has two target functions. Depending on the use case one can perform better than the other and in this case the rotation translation function is best. Also we could dock four copies of the chain into the structure by setting "Copies" to 4 however to make the tutorial faster we will only use 1. Also Molrep removes all water atoms whilst fitting.

Whilst Molrep is running (it will take ~10mins) you can also try and dock the model into the map by hand.

- Customise Coot for cryoEM:
    - `Calculate -> Modules -> Cryo-EM`
        - Load an additional menu item with cryo-EM tools
    - `Calculate -> Scripting -> Python "curlew()"`
        - From there, select and install "Black Box Morph and Fit", "Chain Refine" and "Morph"

- Move the view to the centre of the density and sets the contour level
    - `Cryo-EM -> Go to Map Molecule Middle`
- Aligns the centre of the molecule to the centre of the map
    - `Calculate -> Move Molecule Here`
- Show the structure as a CA ribbon:
    - `Display manager -> homolog.pdb -> change "Bonds…" to "CAs…"`
- Use Rotate/Translate Zone tool to move the chain into close to the correct orientation
    - Select molecule:



- Once close to the position try using the rigid body fitting in order to optimise the overall fit of the model to the map:
    - `Calculate -> Modelling -> Rigid Body Fit Molecule…`
- Does the model now fit the map reasonably well? If yes click "Accept" in the "Accept Refine" panel. If not then "Reject".
- If you are happy with the fit make sure to save the coordinates:
    - `File -> Save Coordinates…`

Note: In practice, if the position or rotation of the model is too far from the optimal solution in the map, some fitting tools may not work. In such cases, you may need to try various tools before achieving a reasonable model from which to start full-model refinement. There are a variety of tools in Coot that can help with this (e.g. Rotate Translate Zone/Chain/Molecule (from the model toolbar, and select Molecule) , Jiggle Fit "J", Morph Fit, real space refinement with self-restraints, etc.).

(c) Inspect differences between the model and map (don't spend too long on this!)

- Use the fitting model from the molrep task (molrep.pdb) the model you fitted model or the fitted model supplied (homolog_molrep.pdb).
- Are there any regions of the model for which there is no supporting density?
    - Validate -> Density fit analysis
- Remove three large regions (consecutive stretches of residues) for which there is no supporting evidence in the map (e.g. see residues 730-735).
    - Right toolbar: Delete Item (Bin Icon) -> Delete Zone -> click two residues

  Remember that this model came from a different (homologous) structure, so it is no surprise that there are differences between the starting model and the map.

(d) After any of the above quick pre-refinement model trimming that you choose to do – i.e. removing loop regions for which there is little or no supporting map density – save the coordinates of the fitted model:

- File -> Save Coordinates

Close Coot and go to the next step using either with your model or with the provided file: homolog_molrep_prepared.pdb

Part 2) Ligand generation

Before being able to refine the model, we must ensure that we have restraint dictionaries for all new/unknown ligands (many common ligands are in the CCP4 Monomer Library and dealt with automatically). Our model contains an unknown ligand "PTP". Consequently, we must create a dictionary for this compound. The provided file "PTP.cif" is an mmCIF file that contains a description that matches the ligand in the model (see the *Note* below). Now create a dictionary for this ligand using AceDRG:

- In the CCP-EM interface, select the AceDRG task and set the following:
    - mmCIF file: PTP.cif
    - Monomer code: PTP

      This will ensure that the output dictionary will work with the PTP ligand in the model.
- Now run the job. It will take a few minutes.

Running this command will create a dictionary called “PTP_acedrg.cif”, along with coordinates for a low-energy conformer “PTP_acedrg.pdb”, in the job directory. Once the job has finished running, select the “Launcher” tab to find the location of these files – you will need this for the next step.

Note: mmCIF files for ligands that aren’t in the CCP4 Monomer Library can often be found on the PDBeChem website (https://www.ebi.ac.uk/pdbe-srv/pdbechem). If an mmCIF file were not available, we could instead have used a SMILES string corresponding to the ligand. This would be an adequate starting point for ligand dictionary generation, the problem being that the atom names would be lost. Consequently, either the atoms would need to be renamed, or alternatively the ligand could be removed from the model and coordinates with the new atomic nomenclature refitted.
Addendum: The ligand used in this tutorial, 2-phenylethyl 1-thio-beta-D-galactopyranoside, is normally given the code PTQ. Since the tutorial was written, PTQ has been added to the CCP4 monomer library and therefore the refinement would run without the need to create a custom ligand dictionary. For this tutorial the ligand has been renamed as PTP to ensure the AceDRG step is still needed.

Part 3) Refinement

`(a) Run refinement`

Now try to refine the model again, using the newly created ligand dictionary.

- In the CCP-EM interface, select the Refmac Servalcat task.
- To scale and calculate difference maps Refmac Servalcat now uses unfiltered and unsharpened half maps and a mask. These should be taken from the final 3D refinement and the mask used for processing in Relion (or similar).

- Masked refinement (previously called local refinement) creates a sub-volume around the atomic model and refines this area only. This speeds up refinement dramatically if you have a partial model and reports statistics for this area only.

- Enter the following parameters:
    - `Input model: your fitted model or homolog_prepared_fitted.pdb`
    - `Restraints dictionary: PTP_acedrg.cif`
    - `Resolution (2.9 Å)`
    - `Half map 1: run_half1_class001_unfil.mrc`
    - `Half map 2: run_half2_class001_unfil.mrc`
    - `Mask for Fo-Fc map: mask.mrc`
    - `Masked refinement: select`
    - `Refinement options -> Refmac cycles: 10`
    - `Refinement options -> Strict symmetry: D2`
- Now run the job (this might take a few minutes, depending on computing power).

`(b) Inspect the refinement statistics`

- Once the job has finished, click on the “Results” tab. Look at the refinement statistics table, as well as the graphs.
- Were 10 cycles sufficient, i.e. has the refinement converged?
- Is there evidence that refinement has improved the model, or made it worse? Consider statistics representing fit to the data (FSC average), as well as geometric quality (Rms bond/angle/chiral).
- What is the major contributing factor to the improvement of refinement statistics between Start and Finish? Is it surprising that the refinement statistics have improved? (hint: are we only refining coordinate positions?)
- Look at the Validation tab in the Results tab, is there significant deviation between the model vs work and free half map FCSs?
- Why are the statistics so good even though only a single chain is present?

`(c) Visual inspection`

Click on the “Coot” button at the top-left of the interface. Coot will open with two maps and both the model before and after refinement loaded and displayed.

Click Display manager and look at the two maps:

- diffmap.mtz FWT PHWT corresponds to the sharpened and weighted full map combined from the input half maps.
- diffmap.mtz DELFWT PHDELWT corresponds to the sharpened and weighted difference map.
    - Hide the difference map for now

- Inspect the models. Can you see any evidence of changes in the model, or improvements to local model quality?
- Zoom out so that you can see the whole model(s). Open the Display Manager. Hide the map, and change the representation of both models to “CAs + Ligands”. Now repeatedly toggle the display of one of the models on and off. What differences can

you perceive between the models? What does this tell you about the differences between the underlying macromolecular structures?
- Change the representation of both models to “Colour by B-factors - CAs”. Now display/undisplay each of the two models in turn, and consider the colouring of the two models. What does this tell you about the differences between the biological assemblies of the two models (i.e. the structure under refinement versus the model of the homologous macromolecule)? Does this reflect anything about the relative resolutions of the maps underlying the two models?
- Compare the Ramachandran plots corresponding to the two models
    - `Validate -> Ramachandran Plot`
- Are any differences due to overall trends or individual residues, and are they substantial or minor?

To see the full biological assembly load refined_expanded.pdb. This file will contain all 4 symmetrically related copies of the betagal chain.

Note: For detailed information on Servalcat’s map calculation please see Yamashita et al. 2021.

Part 4) Optimising Refinement Weight

Now let’s see if we can improve the model by adjusting refinement parameters. We want to improve the fit-to-data (as judged by the FSC), without overly negatively affecting the geometry (agreement with prior knowledge).
The weight used during refinement can be found directly under the “Refinement Statistics” table on the Results page – make a note of what this value was for the previous refinement run (at the time of writing the automatic weight is ~3.25). In order to loosen the geometry / improve fit-to-density, we need to increase this value to increase the weight given to the map restraints with respect to the geometry restraints.

- Clone the previous Refmac5 refinement job in CCP-EM (double click on the job, and then select “Clone” from the top left of the window).
    - `Refinement options -> Auto weight: Unselect`
    - `Refinement options -> Auto weight scale: 16`
        - Specify an Auto weight scale that is 5–10x higher than the auto weight from the previous run (e.g. ~16).
    - `Cross validation: select`
        - This will help to compare the effect of weight
    - Run the job

- Repeat above and run another job with Auto weight scale 5-10x lower (e.g. ~0.65) and also rerun the original job (i.e. Auto weight selected) with cross validation also selected.

Run the above jobs in parallel and whilst these jobs are running you can start the next part.

- Once all three have finished, compare the refinement statistics from this job and the original one.
- Inspect the work and free FSC in the validation tab. How is this affected by the weight?

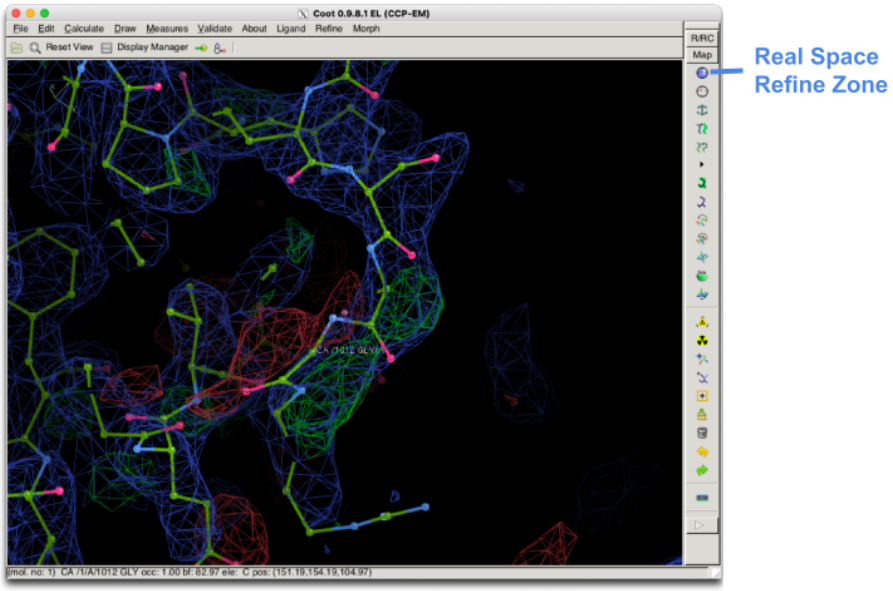
- Click on the “Coot” button in the upper-left portion of the window. Look at the Ramachandran plot from this refinement run, and compare with the original.

Which of the weighting schemes would you use?

Part 5) Manual building and Validation

Unfortunately not all issues can be solved automatically. It is often necessary to inspect and correct the model in Coot in between rounds of refinement.

- From a Refmac Servalcat job window once the job has finished, click the “Coot” button in the upper-left corner of the window.
- Set the density (diffmap.mtz FWT PHWT) and difference maps (diffmap.mtz DELFWT PHDELWT) to appropriate contour levels (in this case ~1.4 and ~2.6 abs).
- Inspect the model and map and see if you can identify any discrepancies between the data and refined model.
- For example, go to residue 1012. The residues here do not fit the data well and the difference map highlights the discrepancy:



- Try using Coot’s Real Refine Zone tool to fix the model
  - Click the tool button and then click two atoms in the structure to define the Zone to be refined.
  - *The real-space refinement tool is near the top of the right-hand toolbar. You will need to select a suitable map for Coot to use for the refinement, and possibly also adjust the refinement weight – see the “R/RC” button in the right-hand toolbar.*

- There are various tools in Coot to help find errors in the model, notably (but not exhaustively):
  - `Validate -> Ramachandran Plot`
    - Look for Outlier residues. You click the residues in the Ramachandran plot and it will orientate the structure (e.g. residue 3). See if you can manually fix these.

    - `Validate -> Density fit analysis`
        - Set the scale and look for regions with high (bad) scores (e.g. residues 771-772. As above, clicking on the chart will orientate the structure. Try manually fixing this.
- The difference map is also a very useful tool use to find regions where the model doesn't agree with the data:
    - Go to residue 1012 what does the difference tell you?
    - Can you use this information to correct the model?

Note: the difference maps are not recalculated as the structure is moved in Coot. To see if the difference map improves, save the coordinates and rerun Refmac to recalculate.

When you have corrected the problems you identified save the modified structure:

- `File -> Save Coordinates`

You can then re-run Refmac using the previously optimised parameters. Does the overall structure quality improve? Again consider statistics representing fit to the data (FSC average), as well as geometric quality (Rms bond/angle/chiral).

Look at the re-refined structures in Coot. Has the difference map improved in the areas you improved? What is the effect on the B-factors in these areas?

Part 6) Adding external restraints

When refining the model, we've been using jelly-body restraints, which are enabled by default in the CCP-EM interface. These are very useful in helping to stabilise refinement, but can't help us to improve the model. Another option is to use restraints from ProSMART in order to inject prior information from a high-resolution homologue during refinement, which will help the model to retain a conformation that is more consistent with prior observations. Specifically, these restraints will ensure that the local interatomic distances within the model do not stray too far from that in the homologous model.

When using ProSMART restraints, it is important to use the best quality reference model that you can. PDB-REDO provides re-refined models corresponding to PDB entries that were determined using X-ray diffraction – this is a useful resource when looking for reference models.
The 1.75 Å model with PDB code 3t09 is identical in sequence to the beta-galactosidase model we have been refining.

- Open a web browser and navigate to PDB-REDO (https://pdb-redo.eu).
- Enter the PDB code 3t09, to get to the relevant model summary.
- Download the PDB file corresponding to the "Re-refined and rebuilt structure".

`(a) Adding ProSMART reference restraints`

- From the main CCP-EM screen, select "ProSMART" from the task list on the left.
    - `Alignment mode: Reference model`
    - `Target PDB(s): homolog_molrep_prepared.pdb (or your model)`
    - `Reference PDB(s): 3t09_final.pdb`

        - Select chain A
- Now run the job.

(a) Refinement with ProSMART restraints

We now need to provide these restraints during refinement, in place of the jelly-body restraints:

- Clone the previous Refmac5 refinement job in CCP-EM.
    - Refmac cycles: 10
        - Note using ProSMART restraints can speed up convergence
    - Auto weight: selected
    - Jelly body: False
        - Note that jelly-body restraints and external restraints work against each other – at present it is best to use one or the other, but not both.
    - Add hydrogens: Ignore
- Click on “External restraints”
    - Use restraints: selected
    - Restraints file: /ProSMART_12/homolog_molrep_prepared.txt (or similar)
        - You should find the ProSMART job directory in your ccpem project
- Run the job.

Once it has finished, compare the refinement statistics from this job and the previous jobs.

Part 6) Further building and Validation in Coot

`(a) Building small sections`

This is an example of model (re)building in Coot and these tools are useful for (re)building small sections of a protein. To demonstrate this, choose a section of the protein to remodel e.g. residues 285-289 (YADRV). In the right-hand toolbar use “Delete Item” (the bin icon), select “Residue/Monomer” and click on the residues to remove (tip - check “Keep Delete Active”). Then select the “Add Residue” button to replace the deleted residues by clicking on the nearest existing residue. Coot will add an alanine by default. Mutate it to the correct residue using the “Mutate and AutoFit” button (upper radiation sign). You may need to adjust the fit using “Real Space Refine Zone”.

`(b) Building large sections - "Baton building"`

This section is based on Paul Emlsey & Paul Bond’s Coot tutorial (https://paulsbond.co.uk/coot-workshop/part2.html)

Sometimes you may be required to build large sections of structure by hand e.g. the resolution is too low for automated *de novo* building, or there is a lack of good homologous structures or predictions. In these cases you can use Coot’s Baton building method.

To test this go to the starting residue MET1A. This allows you to build the complete chain in the correct direction and you can directly compare it to the real structure afterward. Once you are at residue MET1A, use the Display Manager to turn off the display of any existing molecules. Don’t look at it again until you have finished building, validating and refining!

The backbone trace through the density can be more easily visualised by using a skeleton map.

- `Calculate -> Map Skeleton and change it to On.`

To start to “baton build”

- `Calculate / Other Modelling Tools -> C-alpha Baton Mode`

A new menu will be displayed, as well as a white baton and a set of pink points that show possible places for a C-alpha that are 3.8 Å away from the current position. Note that when you start, you are placing a CA at the baton tip for residue 1 (the N-terminus). After placing CA for residue 1, it will switch to building in the C-terminal direction and you will get a choice of positions for residue 2, which is currently at the centre of the view. This might seem that you are “doubling-back” on yourself, which can be confusing the first time, but it is useful for extending existing chains.

Press “Try Another” and “Previous Tip Position” to cycle through the points, Lengthen and Shorten to change the baton length, and Accept to place a CA and move onto the next residue. If none of the guide points are suitable you can use b to toggle baton swivel mode.

Build from the N-terminus to the C-terminus. Try to build the first ~25-50 residues, which will probably take ~10-15 minutes. If you make a mistake you can press Undo to move back one residue. Click Dismiss on the baton controls when done.

Now to turn these CA positions into mainchain:

- `Calculate / Other Modelling Tools -> Ca Zone to Mainchain`

Click on the Baton model. It may take several seconds while it builds (note that you need at least 6 residues for this to work). Two new molecules will be created that are traced in different directions. We know the forward direction is correct (see how much better the carbonyls fit) so the “mainchain-backwards” and Baton Atoms models can be deleted.

To optimise the new chain’s fit-to-density use:

- `Refine -> Chain Refine`

Accept the results if the fit is better.

The next step is to convert the poly-alanine chain into a fully sequenced chain by assigning the sequence. There are two options for this. Option 1 is useful if you know the identity for all the residues you have built. Option 2 can be used if you know the sequence for the whole chain but have only built part of it.

Assigning the sequence - Option 1:

- `Calculate -> Mutate Residue Range`

Assigning the sequence - Option 2:

- `Calculate -> Assign Sequence -> Dock Sequence (py)`

Remember to select the “mainchain-forwards” model to mutate.
The full sequence is:

```
>homolog_prepared_fitted.pdb|Chain=A
MITDSLAVVLQRRDWENPGVTQLNRLAAHPPFASWRNSEEARTDRPSQQLRSLNGEWRFAWFPAPEAVPESWLECDLPEA
DTVVVPSNWQMHGYDAPIYTNVTYPITVNPPFVPTENPTGCYSLTFNVDESWLQEGQTRIIFDGVNSAFHLWCNGRWVGY
GQDSRLPSEFDLSAFLRAGENRLAVMVLRWSDGSYLEDQDMWRMSGIFRDVSLLHKPTTQISDFHVATRFNDDFSRAVLE
AEVQMCGELRDYLRVTVSLWQGETQVASGTAPFGYADRVTLRLNVENPKLWSAEIPNLYRAVVELHTADGTLIEAEACDV
GFREVRIENGLLLLNGKPLLIRGVNRHEHHPLHGQVMDEQTMVQDILLMKQNNFNAVRCSHYPNHPLWYTLCDRYGLYVV
DEANIETHGMVPMNRLTDDPRWLPAMSERVTRMVQRDRNHPSVIIWSLGNESGHGANHSRPVQYEGGGADTTATDIICPM
YRPLILCEYAHAMGNSLGGFAKYWQAFRQYPRLQGGFVWDWVDQSLIKYDENGNPWSAYGGDFGDTPNDRQFCMNGLVFA
DRTPHPALTEAKHQQQFFQFRLSGQTIEVTSEYLFRHSDNELLHWMVALDGKPLASGEVPLDVAPQGKQLIELPELPQPE
SAGQLWLTVRVVQPNATAWSEAGHISAWQQWRLAENLSVTLPAASHAIPHLTTSEMDFCIELGNKRWQFNRQSGFLSQMW
IGDKKQLLTPLRDQFTRAPLDNDIGVSEATRIDPNAWVERWKAAGHYQAEAALLQCTADTLADAVLITTAHAWQHQGKTL
FISRKTYRIDGSGQMAITVDVEVASDTPHPARIGLNCQLAQVAERVNWLGLGPQENYPDRLTAACFDRWDLPLSDMYTPY
VFPSENGLRCGTRELNYGPHQWRGDFQFNISRYSQQQLMETSHRHLLHAEEGTWLNIDGFHMGIGGDDSWSPSVSAEFQL
SAGRYHYQLVWCQK
```

You can then use Coot’s other validation and rebuilding tools to optimise your new structure followed by automated refinement with Refmac. You can also unhide the pre-existing model and see how well your de novo structure agrees with it.

Building models correctly is a time consuming process but it is necessary to give you and any others who may use it in the future the best possible structure to work with.

Refmac Servalcat references:

Yamashita, K., Palmer, C. M., Burnley, T., Murshudov, G. N. Cryo-EM single particle structure refinement and map calculation using Servalcat. *Acta Cryst* D77, 1282-129, 2021.

Current approaches for the fitting and refinement of atomic models into cryo-EM maps using CCP-EM. Nicholls, R.A., Tykac M., Kovalevskiy, O., & Murshudov, G.N. *Acta Cryst* D74, 492-505, 2018.

CCP-EM reference:
Burnley, T., Palmer, C.M. & Winn, M. Recent developments in the CCP-EM software suite. *Acta Cryst* D73, 469-47, 2017.

ProSMART reference:
Nicholls, R.A., Long F. & Murshudov, G.N. Low Resolution Refinement Tools in REFMAC5. Acta Cryst. D68, 404-417, 2012.

PDB_REDO reference & link:
Joosten, R.P. & Vriend G. PDB_REDO: automated re-refinement of X-ray structure models in the PDB. *J. Appl. Cryst*. 42, 376-384, 2009.
https://pdb-redo.eu/

Contact:
Do please report any issues or bugs.... it's much appreciated and helps us make the software better:
ccpem@stfc.ac.uk